

## **REMARKS**

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 13-24 are pending, of which claims 13-24 have been amended.

### **35 U.S.C. §101 Claim Rejections**

Claims 21-24 are rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter (Office Action p.2). Appropriate amendments to claims 21-24 have been provided herein. Accordingly, claims 21-24 are in condition for allowance and Applicant respectfully requests that the §101 rejection be withdrawn.

### **35 U.S.C. §102 Claim Rejections**

Claims 13-24 are rejected under 35 U.S.C. §102(b) as being anticipated by Aho et al., "Compilers: Principles, Techniques, and Tools" (1986) (hereinafter, "Aho"). Applicant respectfully traverses the rejection.

The Office relies on Aho which is a textbook that describes compilers. A compiler reads a program written in a particular source language and translates it into an equivalent program in a target language, such as another programming language or machine code (Aho p.1, §1.1).

The Office states that "Aho teaches techniques to build a compiler" which is a computer program (Office Action p.4) and that this is a general basis to reject the claims. Although Aho describes aspects of compiler design in Chapter 11 of the textbook, the Office only relies on an introduction to compiling in Chapter 1.

1 Specifically, the Office refers to three analysis phases of compiling which include  
2 lexical, syntax, and semantic analysis (Aho p.12, §1.3). The Office recognizes  
3 that a compiler can generate a programming construct into a syntax tree during  
4 syntax analysis (Office Action p.4; Aho p.12, §1.3).

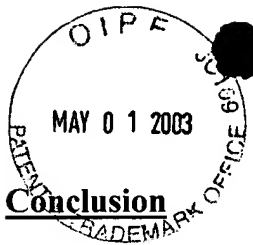
5 The analysis phases of compiling, as well as generating a syntax tree during  
6 compiling, are described in the Background section of the present application (p.1,  
7 line 19 – p.2, line 18). In contrast to a compiler that translates a program written  
8 in a specific programming language into another programming language or  
9 machine code, Applicant describes that a program can be developed and  
10 represented by a “high-level program tree that is a syntax-independent  
11 representation” of a programmer’s intent (Description, p.4, lines 33-37). A  
12 programmer can directly manipulate the program tree, “which is in contrast to  
13 conventional programming systems in which a programmer manipulates a textual  
14 representation of the program that is later converted into a syntax tree during  
15 compilation” – as described in Aho. (Description, p.5, lines 1-5).

16  
17 Claim 13 for example, describes “a first node of a data structure, the first  
18 node representing a syntax-independent programming intent”. Aho does not show  
19 or disclose a node of a data structure “representing a syntax-independent  
20 programming intent”, as recited in claim 13. Aho only describes that a compiler,  
21 which is language specific, generates a syntax tree from a token stream that  
22 represents a programming statement during the compiling process (Aho p.12,  
23 §1.3). Accordingly, claim 13, as well as dependent claims 14-16, are allowable  
24  
25

1 over Aho and Applicant respectfully requests that the §102 rejection be  
2 withdrawn.

3  
4 Claim 17 recites a method of handling data comprising “reading a first node  
5 of a hierarchical tree, the first node representing a syntax-independent  
6 programming intent”. Aho does not show or disclose a node of a hierarchical tree  
7 “representing a syntax-independent programming intent”, as recited in claim 17.  
8 As described above in the response to the rejection of claim 13, Aho describes that  
9 a compiler generates a syntax tree from a token stream that represents a  
10 programming statement during the compiling process (Aho p.12, §1.3).  
11 Accordingly, claim 17, as well as dependent claims 18-20, are allowable over Aho  
12 and Applicant respectfully requests that the §102 rejection be withdrawn.

13  
14 Claim 21 recites a data structure comprising “a first node representative of  
15 a syntax-independent programming intent”. Aho does not show or disclose a node  
16 of a data structure that is “representative of a syntax-independent programming  
17 intent”, as recited in claim 21. As described above in the response to the rejection  
18 of claim 13, Aho describes that a compiler generates a syntax tree from a token  
19 stream that represents a programming statement during the compiling process  
20 (Aho p.12, §1.3). Accordingly, claim 21, as well as dependent claims 22-24, are  
21 allowable over Aho and Applicant respectfully requests that the §102 rejection be  
22 withdrawn.

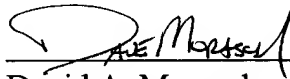


Conclusion

Pending claims 13-24 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. If any issues remain that preclude issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Dated: May 1, 2003

By:   
David A. Morasch  
Reg. No. 42,905  
(509) 324-9256 x 210

RECEIVED

MAY 05 2003

Technology Center 2100